

## Softwarepraktikum

# **Business Rules in Semantic Wikis**

Sebastian Furth  
Alexander Legler  
Florian Ziegler

4. März 2010

# Inhaltsverzeichnis

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Business Rules</b>                                | <b>2</b>  |
| <b>2</b> | <b>Kollaboratives Knowledge Engineering in Wikis</b> | <b>3</b>  |
| 2.1      | Klassisches und kollaboratives KE . . . . .          | 3         |
| 2.2      | Möglichkeiten zum kollaborativen KE . . . . .        | 3         |
| 2.3      | Was ist KnowWE? . . . . .                            | 3         |
| <b>3</b> | <b>Technische Umsetzung</b>                          | <b>5</b>  |
| 3.1      | Technische Grundlagen von KnowWE . . . . .           | 5         |
| 3.2      | Terminologie im DroolsPlugin . . . . .               | 5         |
| 3.2.1    | Inputs . . . . .                                     | 5         |
| 3.2.2    | Values . . . . .                                     | 7         |
| 3.3      | Wiki-Markup . . . . .                                | 7         |
| 3.3.1    | DroolsFactsSection . . . . .                         | 7         |
| 3.3.2    | DroolsRulesSection . . . . .                         | 8         |
| 3.3.3    | DroolsSessionSection . . . . .                       | 9         |
| 3.4      | Die Drools Shell . . . . .                           | 10        |
| 3.4.1    | Benutzung . . . . .                                  | 12        |
| <b>A</b> | <b>Gesamte UML-Diagramme</b>                         | <b>13</b> |
| A.1      | OOA-Model . . . . .                                  | 13        |
| A.2      | OOD-Model . . . . .                                  | 14        |
| <b>B</b> | <b>Beispiel: Car Diagnosis</b>                       | <b>15</b> |

# Kapitel 1

## Business Rules

Eine **Business Rule** ist ein Ausdruck, der zur Problemlösung innerhalb eines Prozesses oder einer abgegrenzten Aufgabenstellung beitragen kann. Business Rules können sowohl in Form von Beschreibungen als auch von Einschränkungen auftreten.

Es gibt eine Vielzahl von Business Rule Management Systemen (BRMS). Zu den Bekanntesten gehören JESS sowie das im Rahmen dieser Arbeit betrachtete JBoss Drools.

In diesem Bericht wird die Integration von JBoss Drools in ein bestehendes Wissensmanagementsystem in Form eines Semantic Web Wikis beschrieben.

# Kapitel 2

## Kollaboratives Knowledge Engineering in Wikis

### 2.1 Klassisches und kollaboratives KE

Das Knowledge Engineering sollte insbesondere in seiner klassischen und der kollaborativen Form unterschieden werden.

Das klassische Knowledge Engineering baut vorallem auf das Expertenwissen einer begrenzten Anzahl von Personen (Community of Practice) und den methodischen Fähigkeiten eines Wissensingenieurs hinsichtlich Formalisierungsaspekten.

Beim kollaborativen Knowledge Engineering steht der Gedanke des verteilten Wissens im Vordergrund. Hierbei sollte besonders an Web-Communities gedacht werden, welche einen großen Wissensbestand, allerdings keine geeignete Plattform zu deren gezielter Verwaltung besitzen.

### 2.2 Möglichkeiten zum kollaborativen KE

Zur Wissensakquise in kollaborativen Umgebungen eignen sich unterschiedlichste Methoden, wie z.B. Web-Portale, Foren und Wikis. Wie am Beispiel der freien Enzyklopädie Wikipedia zu erkennen ist, eignen sich Wikis hervorragend zur Sammlung von Wissen. Da dieses Wissen allerdings nur informell vorliegt, bedarf es einer Möglichkeit zur Formalisierung. Diese Lücke wird durch das Semantic Web Wiki KnowWE geschlossen.

### 2.3 Was ist KnowWE?

KnowWE (Knowledge Wiki Environment) ist „[e]in Semantisches Wiki, welches auf dem open-source Wiki JSPWiki aufbaut. Problemlösungswissen kann direkt im Wiki editiert und

ausgeführt werden. Die im Wiki entwickelten Wissensbasen können exportiert und beispielsweise in OEM oder embedded Anwendungen verwendet werden. Weiterhin kann das Wissen über die Ontologiesprache OWL ausgetauscht werden.“<sup>1</sup>

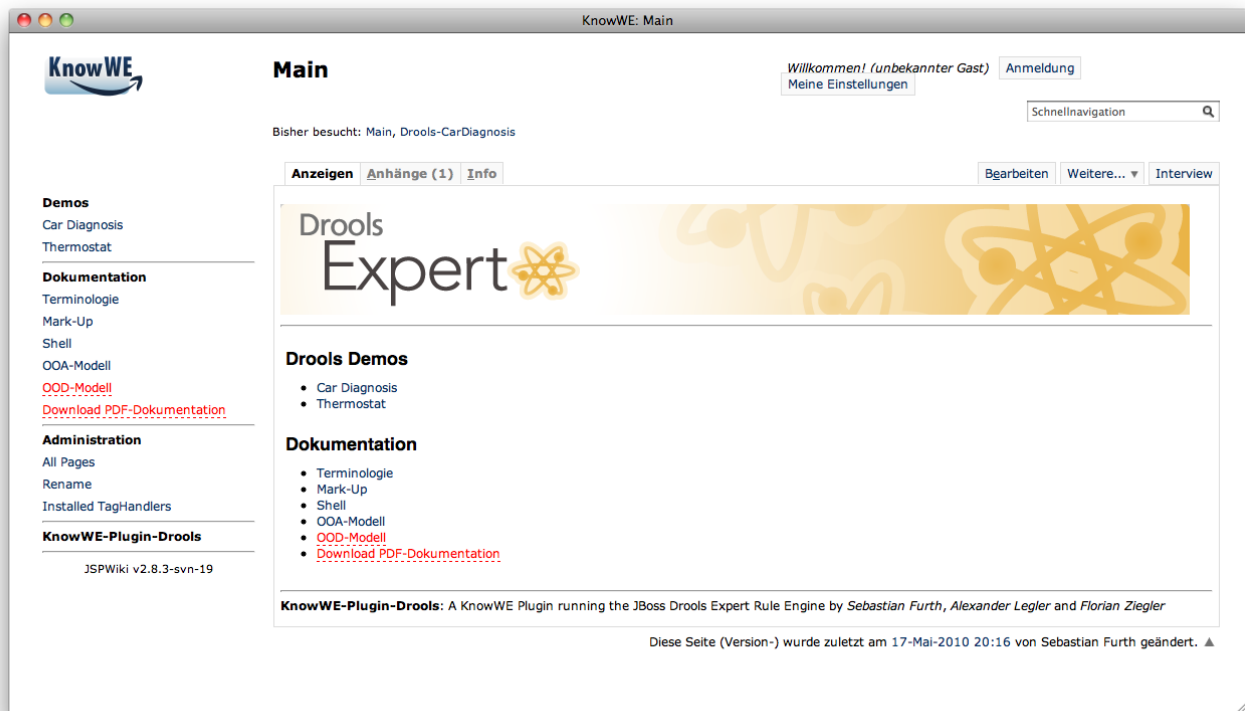


Abbildung 2.1: Screenshot des Interfaces von KnowWE

<sup>1</sup>Nach: Seite „D3web“. In: Wikipedia, Die freie Enzyklopädie.

# Kapitel 3

## Technische Umsetzung

### 3.1 Technische Grundlagen von KnowWE

KnowWE basiert auf drei Grundkonzepten: **ObjectTypes**, **Taghandler** und **Actions**.

ObjectTypes übernehmen die Kernaufgabe der Wissensformalisierung. Daneben werden Taghandler insbesondere zum Rendering von Benutzerinteraktionselementen verwendet. Auf das Konzept der Actions wird zurückgegriffen wenn Benutzereingaben verarbeitet werden sollen.

### 3.2 Terminologie im DroolsPlugin

Die Terminologie im DroolsPlugin fußt auf zwei Kernkonzepten: **Inputs** und **Values**, wie in Abbildung 3.1 gezeigt.

Inputs kapseln zumeist die im System vorhandenen Fragen und bieten je nach Ausprägung verschiedene Möglichkeiten für die Beantwortung an.

Values sind Objektcontainer, die zum Setzen von Werten in Inputs verwendet werden.

#### 3.2.1 Inputs

Bei den Inputs sind `ChoiceInput`, `NumInput` sowie `SolutionInput` zu unterscheiden.

##### ChoiceInput

Ein `ChoiceInput` lässt sich nach der Anzahl von Antwortmöglichkeiten in `MultipleChoiceInput` und `OneChoiceInput` klassifizieren. Die Anzahl der Antwortmöglichkeiten definiert wieviele Values gleichzeitig in einem `ChoiceInput` gesetzt sein dürfen. Die Antwortmöglichkeiten

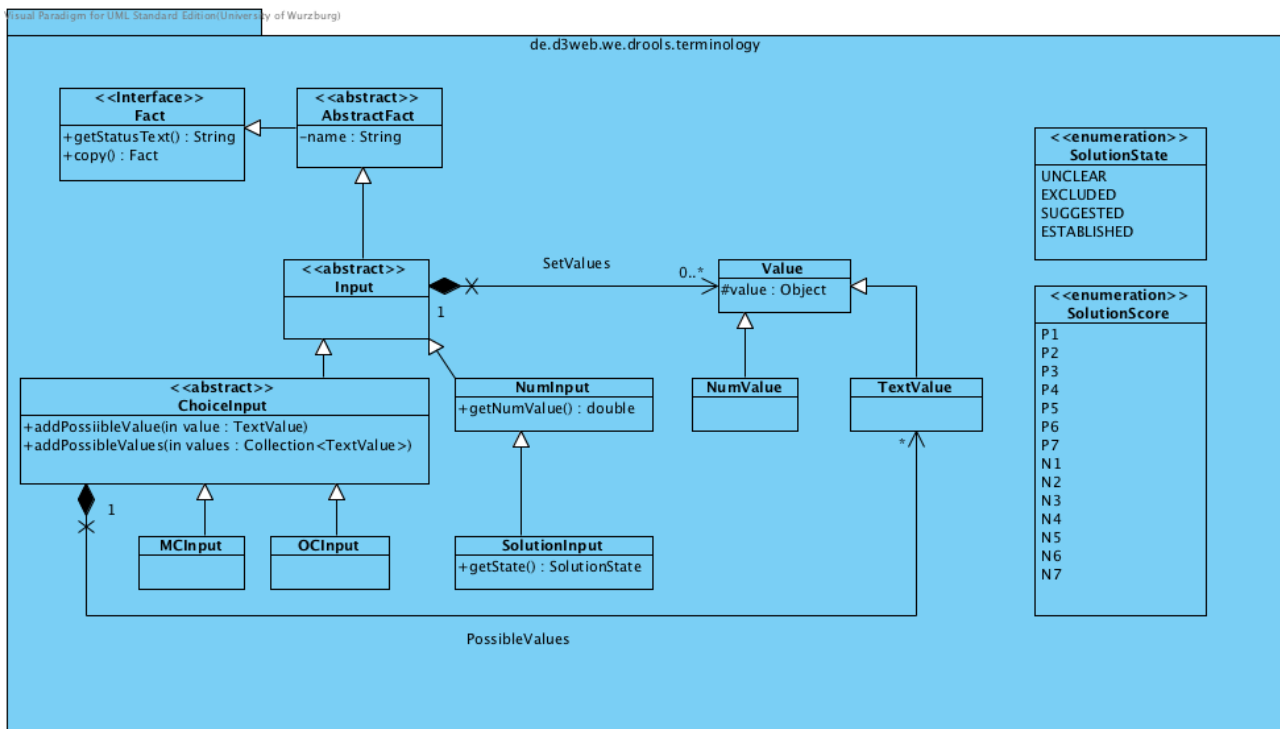


Abbildung 3.1: UML-Klassendiagramm des Pakets `d3.d3web.we.drools.terminology`

sind außerdem durch die `PossibleValues` eingegrenzt, welche eine dem `ChoiceInput` fest zugeordnete Menge von `Values` darstellt.

## NumInput

Ein `NumInput` kann nur einen einzigen numerischen Wert in Form eines `NumValue` aufnehmen.

## SolutionInput

Ein `SolutionInput` stellt eine besondere Form von `NumInput` dar und ist um eine `getState()`-Methode ergänzt, welche eine textuelle Repräsentation des momentanen numerischen Werts in Form eines `SolutionState` zurückgibt. Der numerische Wert der `SolutionInput` ist das Ergebnis des heuristischen Problemlösungsmechanismus, der mithilfe der Drools-Regeln integriert wurde.

Der Problemlösungsmechanismus beruht dabei auf den Werten der folgenden Tabelle:

| Score | N7       | N6  | N5  | N4  | N3      | N2 | N1 | P1 | P2        | P3 | P4 | P5          | P6 | P7  |
|-------|----------|-----|-----|-----|---------|----|----|----|-----------|----|----|-------------|----|-----|
| Wert  | -999     | -80 | -40 | -20 | -10     | -5 | -2 | 2  | 5         | 10 | 20 | 40          | 80 | 999 |
| State | EXCLUDED |     |     |     | UNCLEAR |    |    |    | SUGGESTED |    |    | ESTABLISHED |    |     |

Tabelle 3.1: Übersicht über vorgegebene Scores

Abhängig von der Regel werden dem Score Punkte hinzugefügt oder abgezogen. Der State ergibt sich aus der momentanen Gesamtpunktzahl. Initial ist der Score eines `SolutionInputs` 0.

### 3.2.2 Values

Das `DroolsPlugin` definiert die `NumValue` und `TextValue`-Typen. Ein `TextValue` nimmt einen beliebigen String-Inhalt auf und kann einem `ChoiceInput` zugeordnet werden. Der `NumValue` hat zwei Funktionen: Er kapselt einen `double`-Wert zum einen für die Verwendung mit einem `NumInput` und zugleich stellt er die Scoring-Grundlage für die heuristische Problemlösung im `SolutionInput` dar.

## 3.3 Wiki-Markup

Für das Markup werden drei Sections definiert, die **`DroolsFactsSection`**, welche die Terminologie beinhaltet, die **`DroolsRulesSection`**, welche das Regelwissen bereitstellt sowie die **`DroolsSessionSection`**, welche Befehle speichern und nacheinander abarbeiten kann.

### 3.3.1 DroolsFactsSection

Die Terminologie wird in der `DroolsFactsSection` durch typisierte `Input`-Statements definiert. Gültige Input-Typen sind `OC`, `MC`, `Num` und `Solution`, die im Folgenden erklärt werden. Sämtliche Objekte werden dem *working memory* von Drools zugeführt.

Listing 3.1: Beispiel `DroolsFactsSection`

```
1 %%DroolsFacts
2 Input<OC>("Starter", {"does not turn over", "turns over"});
3 Input<MC>("Driving", {"insufficient power on partial load",
    "insufficient power on full load", "unsteady idle speed", "no
    /else"});
4 Input<Num>("Year of construction");
5 Input<Solution>("Bad ignition timing");
6 Input<Solution>("Flat battery");
7 %
```

### One Choice Inputs

Der One Choice Input stellt eine Einfachauswahlmöglichkeit bereit. Hier wird ein `OC`-Input `Starter` definiert, welcher zwei mögliche Antwortalternativen bietet: `does not turn over` und `turns over`.

```
1 Input<OC>("Starter", {"does not turn over", "turns over"});
```



## Multiple Choice Inputs

Der Multiple Choice Input erlaubt es, mehrere Antwortalternativen auszuwählen. Im Beispiel ist `Driving` ein MC-Input mit folgenden Antwortmöglichkeiten: `insufficient power on partial load`, `insufficient power on full load`, `unsteady idle speed` sowie `no /else`.

```
1  Input<MC>("Driving", {"insufficient power on partial load",
    "insufficient power on full load", "unsteady idle speed", "no
    /else"});
```

## Numerical Inputs

Bei `Year of Construction` handelt es sich um einen numerischen Input, der keine Vorgabewerte hat.

```
1  Input<Num>("Year of construction");
```

## Solution Inputs

`Bad ignition timing` und `Flat battery` stellen Lösungs-Inputs dar, d.h. sie stellen Lösungen wiederum in Input-Form dar, um eine Abstraktionsmöglichkeit innerhalb des Systems zu gewährleisten.

```
1  Input<Solution>("Bad ignition timing");
2  Input<Solution>("Flat battery");
```

## 3.3.2 DroolsRulesSection

In der `DroolsRulesSection` wird das Regelwissen formalisiert. Hierbei wird das Format der DRL, der Drools Rule Language verwendet.

Listing 3.2: Beispiel `DroolsRulesSection`

```
1  %%DroolsRules
2  rule "Flat battery"
3      when
4          $value : Value(value == "does not turn over")
5          Input(name == "Starter" && values contains $value)
6          $solution : SolutionInput(name == "Flat battery")
7      then
8          $solution.setValue(P5);
9      end
10
11 rule "Exclusion of bad ignition timing"
12     when
13         $value : Value(value == "turns over")
```

```

14     not Input(name == "Starter" && values contains $value)
15     $solution : SolutionInput(name == "Bad ignition timing")
16     then
17         $solution.setValue(N7);
18     end
19
20     %

```

Jede Regel besteht aus einem Titel („Flat battery“), einem Bedingungsteil (`when...`), sowie einem Aktionsteil (`then...`).

Grundsätzlich ist bei der Formulierung der Regeln zu beachten, dass die Fact-Bezeichner *case-sensitive* sind.

Im Bedingungsteil werden die Vorbedingungen formuliert, welche erfüllt sein müssen, damit diese Regel feuern kann. Ferner werden Variablen initialisiert, welche im Aktionsteil benötigt werden.

In der Beispielregel „Flat battery“ wird zunächst der Value `does not turn over` aus dem working memory von Drools abgerufen und überprüft, ob der Input mit der Bezeichnung `Starter` auf diesen Value gesetzt ist. Anschließend wird die Variable `$solution` initialisiert, welche für den Aktionsteil benötigt wird.

Im Aktionsteil wird der zuvor initialisierten `$solution`-Variable der Wert `P5` hinzugefügt.

### 3.3.3 DroolsSessionSection

Zum schnelleren Zugriff auf Input/Value-Kombinationen bietet das Drools Plugin die Möglichkeit, sogenannte **Sessions** zu laden und zu speichern. Mittels der Kommandos `store` und `load`, die weiter unten beschrieben werden, können Sessions gespeichert und geladen werden.

Das Drools Plugin serialisiert die Session im folgenden Format:

Listing 3.3: Beispiel DroolsSessionSection

```

1 %%DroolsSession
2 get Starter
3 set Starter = does not turn over
4 fire
5 get Flat battery
6
7 @Name:Foo
8 %

```

Beim Laden der Session werden die gespeicherten Befehle von oben nach unten abgearbeitet. Anzumerken ist, dass Sessions auch manuell im Wiki-Artikel angelegt werden können.

## 3.4 Die Drools Shell

Die Drools Shell ist das zentrale Interaktionselement des DroolsPlugins. Über sie werden sämtliche Eingaben und Ausgaben getätigt.

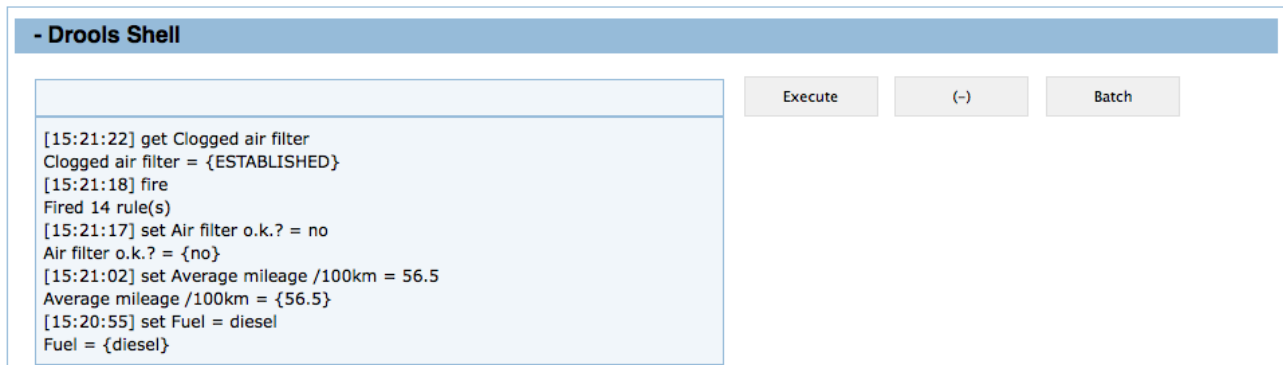


Abbildung 3.2: Screenshot der Drools Shell mit Benutzersession

Der technische Aufbau wird anhand des UML-Diagramms in Abbildung 3.3 erklärt.

Ausgangspunkt bildet der `DroolsTagHandler` welcher ein HTML-Formular mit Textfeld sowie drei Buttons „Execute“, „-/+“ und „Batch“ in die Wiki-Seite rendert. Das Formular kann über den „Execute“-Button oder alternativ per Tastendruck auf <Enter> abgeschickt werden. Der „-/+“-Button blendet die History der Shell aus bzw. wieder ein. Ein Klick auf den „Batch“-Button führt dazu, dass sich die Eingabezeile in ein Eingabefeld verwandelt, in welches mehrere Befehle zeilenweise eingegeben werden können. Zu beachten ist hierbei, dass sich das Formular nicht mehr mit <Enter> abschicken lässt, sondern <Shift> + <Enter> gedrückt werden müssen.

Die vom Benutzer eingegebenen Daten werden zunächst von den hinterlegten JavaScript-Funktionen entgegengenommen und anschließend per AJAX<sup>1</sup> an die `DroolsAction` und `DroolsCodeCompletionAction` gesendet und dort verarbeitet. Die `DroolsAction` übernimmt dabei die Interaktion mit der Terminologie und den hinterlegten Regeln. Auf die Funktionsweise der `DroolsCodeCompletionAction` wird in einem separaten Punkt genauer eingegangen.

Das Ergebnis der Verarbeitung wird KnowWE-typisch über die `KnowWE.jsp` an den Client ausgegeben. Von dort werden sie dann wieder per Javascript verarbeitet und auf der Konsole ausgegeben.

---

<sup>1</sup>Asynchronous JavaScript and XML

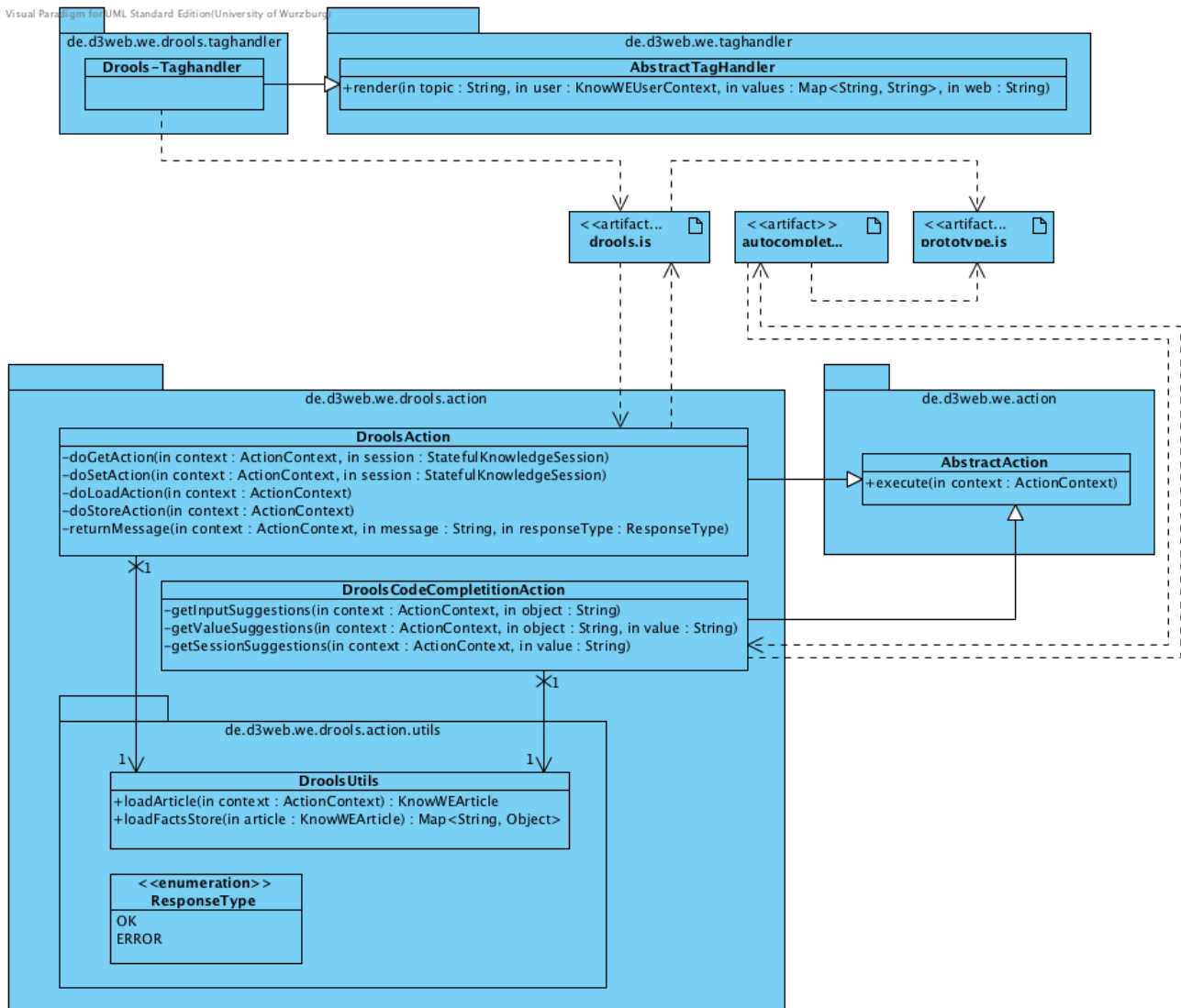


Abbildung 3.3: UML-Klassendiagramm der Drools Shell-Implementierung

### 3.4.1 Benutzung

#### Verfügbare Kommandos

| Kommando              | Erklärung  |
|-----------------------|--|
| help                  | Gibt alle verfügbaren Befehle auf der Drools Shell aus.                                  |
| get <Input>           | Gibt den aktuellen Wert (Value) eines Inputs aus.  |
| set <Input> = <Value> | Setzt den Wert eines Inputs auf den übergebenen Value.                                   |
| fire                  | Feuert alle passenden Regeln.  |
| reset                 | Setzt alle Inputs auf ihre Ausgangswerte zurück.   |
| clear                 | Leert die Konsole.   |
| store <Name>          | Speichert alle zuvor eingegebene Befehle unter dem eingegebenen Namen als DroolsSession. |
| load <Name>           | Lädt die zuvor gespeicherte DroolsSession <Name> in die aktuelle Sitzung.                |

#### Autocompletion

Zur einfachen Bedienung der Drools-Shell wird der Benutzer durch die DroolsCodeCompletionAction bei seinen Eingaben unterstützt. Dies geschieht im wesentlichen dadurch, dass zu seinen bisherigen Eingaben Vorschläge unterhalb der Konsole eingeblendet werden.

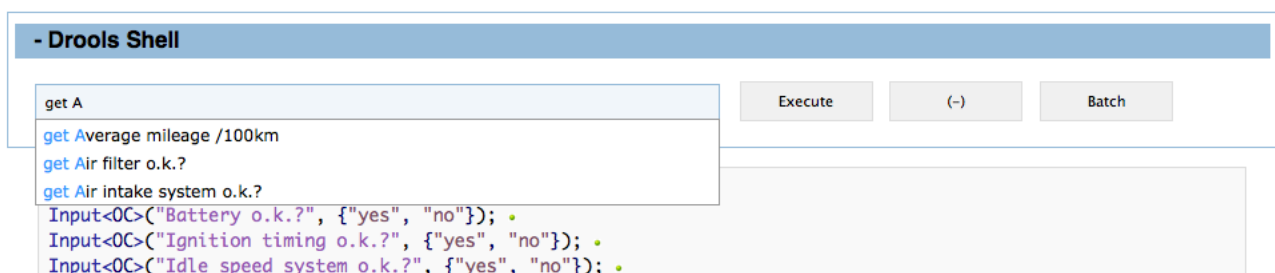


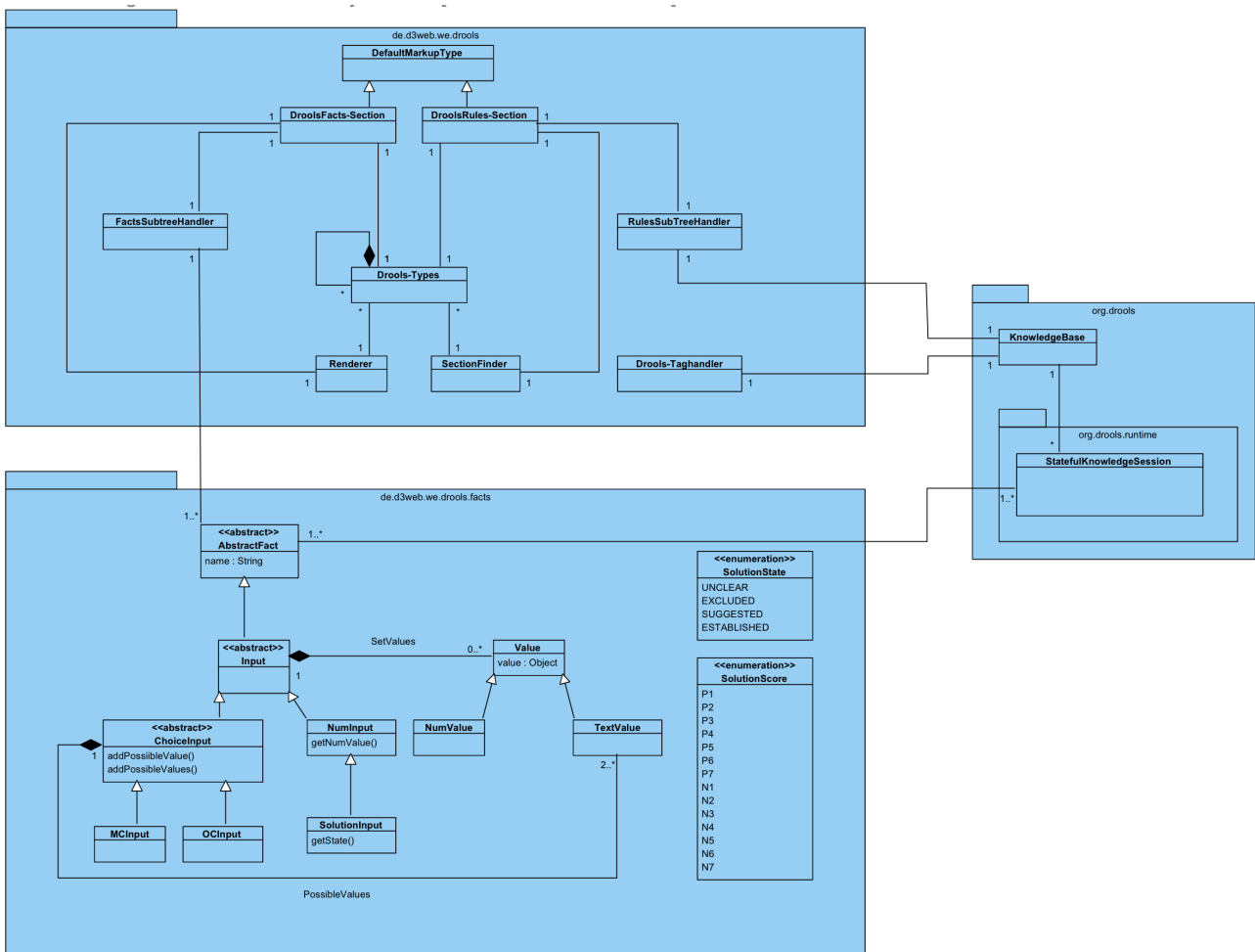
Abbildung 3.4: Screenshot der Drools Shell mit Autocompletion-Feature

Wie dem obigen Beispiel zu entnehmen ist, wird die Benutzereingabe „get a“ durch drei Vorschläge unterstützt. Die Navigation innerhalb der Vorschläge erfolgt über die Pfeiltasten.

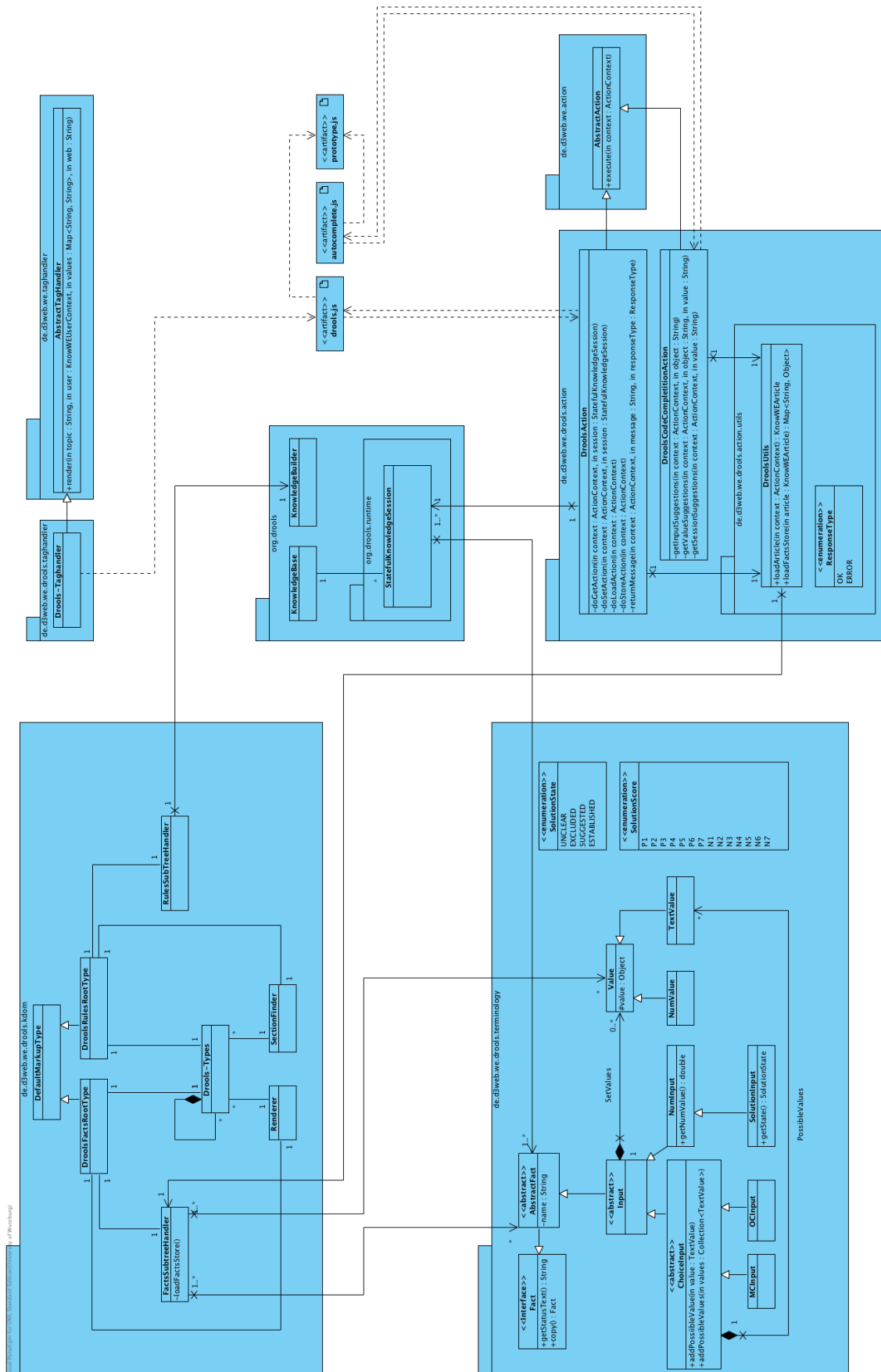
# Anhang A

## Gesamte UML-Diagramme

### A.1 OOA-Model



Visual Paradigm for UML Standard Edition (University of Würzburg)



# Anhang B

## Beispiel: Car Diagnosis

```
1  !!! Drools Demo - Car Diagnosis
2
3  [{KnowWEPlugin drools}]
4
5  %%DroolsFacts
6  Input<OC>("Battery o.k.", {"yes", "no"});
7  Input<OC>("Ignition timing o.k.", {"yes", "no"});
8  Input<OC>("Idle speed system o.k.", {"yes", "no"});
9  Input<Num>("Year of construction");
10 Input<OC>("Air filter o.k.", {"yes", "no"});
11 Input<OC>("Air intake system o.k.", {"yes", "no"});
12 Input<OC>("Starter", {"does not turn over", "turns over"});
13 Input<OC>("Engine start", {"engine barely starts", "engine starts",
    "does not start"});
14 Input<OC>("Make of car", {"VW", "Opel", "Mercedes Benz", "BMW",
    "Porsche", "Fiat", "Toyota", "Mazda", "Other"});
15 Input<MC>("Driving", {"insufficient power on partial load",
    "insufficient power on full load", "unsteady idle speed", "low idle
    speed", "delayed take-off", "weak acceleration", "no /else"});
16 Input<Num>("Average mileage /100km");
17 Input<Num>("Num. Mileage evaluation");
18 Input<OC>("Mileage evaluation", {"slightly increased", "normal",
    "increased"});
19 Input<Num>("Real mileage /100km");
20 Input<OC>("Exhaust fumes", {"black", "blue", "invisible"});
21 Input<OC>("Exhaust pipe color", {"brown", "grey", "light grey", "sooty
    black"});
22 Input<OC>("Exhaust pipe color evaluation", {"abnormal", "normal"});
23 Input<OC>("Fuel", {"diesel", "unleaded gasoline"});
24 Input<OC>("Engine noises", {"knocking", "ringing", "no /else"});
25 Input<Solution>("Damaged idle speed system");
26 Input<Solution>("Leaking air intake system");
27 Input<Solution>("Clogged air filter");
28 Input<Solution>("Bad ignition timing");
29 Input<Solution>("Empty battery");
30
```



```

31  %
32
33  %%DroolsRules
34  rule "R7"
35      when
36          // This rule was converted manually
37          $real : Input(name == "Real mileage /100km" && numValue != 0)
38          $average : Input(name == "Average mileage /100km" && numValue >
              0.0)
39          $input : NumInput(name == "Num. Mileage evaluation")
40      then
41          $input.setValue($real.getNumValue() / $average.getNumValue() *
              100);
42  end
43
44  rule "R51"
45      when
46          $value : Value(value == "no")
47          Input(name == "Battery o.k.?" && values contains $value)
48          $solution : SolutionInput(name == "Empty battery")
49      then
50          $solution.setValue(P7);
51  end
52
53  rule "R52"
54      when
55          $value : Value(value == "yes")
56          Input(name == "Battery o.k.?" && values contains $value)
57          $solution : SolutionInput(name == "Empty battery")
58      then
59          $solution.setValue(N7);
60  end
61
62  rule "R53"
63      when
64          $value : Value(value == "does not turn over")
65          Input(name == "Starter" && values contains $value)
66          $solution : SolutionInput(name == "Empty battery")
67      then
68          $solution.setValue(P5);
69  end
70
71  rule "R54"
72      when
73          $value : Value(value == "turns over")
74          Input(name == "Starter" && values contains $value)
75          $solution : SolutionInput(name == "Empty battery")
76      then
77          $solution.setValue(N4);
78  end
79

```

```

80  rule "R55_1_2"
81    when
82      $value1 : Value(value == "engine barely starts")
83      Input(name == "Engine start" && values contains $value1)
84      $solution : SolutionInput(name == "Empty battery")
85    then
86      $solution.setValue(P5);
87  end
88
89  rule "R55_2_2"
90    when
91      $value2 : Value(value == "does not start")
92      Input(name == "Engine start" && values contains $value2)
93      $solution : SolutionInput(name == "Empty battery")
94    then
95      $solution.setValue(P5);
96  end
97
98  rule "R56"
99    when
100      $value : Value(value == "engine starts")
101      Input(name == "Engine start" && values contains $value)
102      $solution : SolutionInput(name == "Empty battery")
103    then
104      $solution.setValue(N5);
105  end
106
107  rule "R23"
108    when
109      $value : Value(value == "yes")
110      Input(name == "Air intake system o.k.?" && values contains $value)
111      $solution : SolutionInput(name == "Leaking air intake system")
112    then
113      $solution.setValue(N7);
114  end
115
116  rule "R24"
117    when
118      $value : Value(value == "no")
119      Input(name == "Air intake system o.k.?" && values contains $value)
120      $solution : SolutionInput(name == "Leaking air intake system")
121    then
122      $solution.setValue(P7);
123  end
124
125  rule "R16"
126    when
127      $value1 : Value(value == "insufficient power on full load")
128      Input(name == "Driving" && values.size > 0 && values not contains
        $value1)
129      $value3 : Value(value == "unsteady idle speed")

```

```

130     Input(name == "Driving" && values.size > 0 && values not contains
        $value3)
131     $value4 : Value(value == "insufficient power on partial load")
132     Input(name == "Driving" && values.size > 0 && values not contains
        $value4)
133     $solution : SolutionInput(name == "Leaking air intake system")
134     then
135         $solution.setValue(N3);
136 end
137
138 rule "R17"
139     when
140         $value : Value(value == "insufficient power on full load")
141         Input(name == "Driving" && values contains $value)
142         $solution : SolutionInput(name == "Leaking air intake system")
143     then
144         $solution.setValue(P5);
145 end
146
147 rule "R18"
148     when
149         $value : Value(value == "insufficient power on partial load")
150         Input(name == "Driving" && values contains $value)
151         $solution : SolutionInput(name == "Leaking air intake system")
152     then
153         $solution.setValue(P3);
154 end
155
156 rule "R19"
157     when
158         $value : Value(value == "unsteady idle speed")
159         Input(name == "Driving" && values contains $value)
160         $solution : SolutionInput(name == "Leaking air intake system")
161     then
162         $solution.setValue(P1);
163 end
164
165 rule "R20"
166     when
167         $value : Value(value == "slightly increased")
168         Input(name == "Mileage evaluation" && values contains $value)
169         $solution : SolutionInput(name == "Leaking air intake system")
170     then
171         $solution.setValue(P3);
172 end
173
174 rule "R21"
175     when
176         $value : Value(value == "normal")
177         Input(name == "Mileage evaluation" && values contains $value)
178         $solution : SolutionInput(name == "Leaking air intake system")

```

```

179     then
180         $solution.setValue(N4);
181     end
182
183     rule "R22"
184         when
185             $value : Value(value == "increased")
186             Input(name == "Mileage evaluation" && values contains $value)
187             $solution : SolutionInput(name == "Leaking air intake system")
188         then
189             $solution.setValue(P4);
190         end
191
192     rule "R10"
193         when
194             $value : Value(value == "engine barely starts")
195             Input(name == "Engine start" && values.size > 0 && values not
196                 contains $value)
197             $solution : SolutionInput(name == "Damaged idle speed system")
198         then
199             $solution.setValue(N5);
200         end
201
202     rule "R11"
203         when
204             $value : Value(value == "engine barely starts")
205             Input(name == "Engine start" && values contains $value)
206             $solution : SolutionInput(name == "Damaged idle speed system")
207         then
208             $solution.setValue(P5);
209         end
210
211     rule "R12"
212         when
213             $value : Value(value == "unsteady idle speed")
214             Input(name == "Driving" && values.size > 0 && values not contains
215                 $value)
216             $solution : SolutionInput(name == "Damaged idle speed system")
217         then
218             $solution.setValue(N1);
219         end
220
221     rule "R13"
222         when
223             $value : Value(value == "unsteady idle speed")
224             Input(name == "Driving" && values contains $value)
225             $solution : SolutionInput(name == "Damaged idle speed system")
226         then
227             $solution.setValue(P1);
228         end

```

```

228 rule "R14"
229     when
230         $value : Value(value == "low idle speed")
231         Input(name == "Driving" && values.size > 0 && values not contains
                $value)
232         $solution : SolutionInput(name == "Damaged idle speed system")
233     then
234         $solution.setValue(N4);
235 end
236
237 rule "R15"
238     when
239         $value : Value(value == "low idle speed")
240         Input(name == "Driving" && values contains $value)
241         $solution : SolutionInput(name == "Damaged idle speed system")
242     then
243         $solution.setValue(P4);
244 end
245
246 rule "R8"
247     when
248         $value : Value(value == "yes")
249         Input(name == "Idle speed system o.k.?" && values contains $value)
250         $solution : SolutionInput(name == "Damaged idle speed system")
251     then
252         $solution.setValue(N7);
253 end
254
255 rule "R9"
256     when
257         $value : Value(value == "no")
258         Input(name == "Idle speed system o.k.?" && values contains $value)
259         $solution : SolutionInput(name == "Damaged idle speed system")
260     then
261         $solution.setValue(P7);
262 end
263
264 rule "R31"
265     when
266         $value : Value(value == "abnormal")
267         Input(name == "Exhaust pipe color evaluation" && values contains
                $value)
268         $solution : SolutionInput(name == "Clogged air filter")
269     then
270         $solution.setValue(P5);
271 end
272
273 rule "R32"
274     when
275         $value : Value(value == "normal")
276         Input(name == "Exhaust pipe color evaluation" && values contains

```

```

        $value)
277     $solution : SolutionInput(name == "Clogged air filter")
278   then
279     $solution.setValue(N5);
280 end
281
282 rule "R38"
283   when
284     $value1 : Value(value == "engine barely starts")
285     Input(name == "Engine start" && values.size > 0 && values not
        contains $value1)
286     $value2 : Value(value == "does not start")
287     Input(name == "Engine start" && values.size > 0 && values not
        contains $value2)
288     $solution : SolutionInput(name == "Clogged air filter")
289   then
290     $solution.setValue(N4);
291 end
292
293 rule "R39_3_2"
294   when
295     $value3 : Value(value == "engine barely starts")
296     Input(name == "Engine start" && values contains $value3)
297     $solution : SolutionInput(name == "Clogged air filter")
298   then
299     $solution.setValue(P4);
300 end
301
302 rule "R39_4_2"
303   when
304     $value4 : Value(value == "does not start")
305     Input(name == "Engine start" && values contains $value4)
306     $solution : SolutionInput(name == "Clogged air filter")
307   then
308     $solution.setValue(P4);
309 end
310
311 rule "R40"
312   when
313     $value1 : Value(value == "engine barely starts")
314     Input(name == "Engine start" && values.size > 0 && values not
        contains $value1)
315     $value2 : Value(value == "does not start")
316     Input(name == "Engine start" && values.size > 0 && values not
        contains $value2)
317     $solution : SolutionInput(name == "Bad ignition timing")
318   then
319     $solution.setValue(N5);
320 end
321
322 rule "R41_3_2"

```

```

323     when
324         $value3 : Value(value == "engine barely starts")
325         Input(name == "Engine start" && values contains $value3)
326         $solution : SolutionInput(name == "Bad ignition timing")
327     then
328         $solution.setValue(P5);
329 end
330
331 rule "R41_4_2"
332     when
333         $value4 : Value(value == "does not start")
334         Input(name == "Engine start" && values contains $value4)
335         $solution : SolutionInput(name == "Bad ignition timing")
336     then
337         $solution.setValue(P5);
338 end
339
340 rule "R33"
341     when
342         $value1 : Value(value == "unleaded gasoline")
343         Input(name == "Fuel" && values contains $value1)
344         $value2 : Value(value == "black")
345         Input(name == "Exhaust fumes" && values contains $value2)
346         $solution : SolutionInput(name == "Clogged air filter")
347     then
348         $solution.setValue(P5);
349 end
350
351 rule "R25"
352     when
353         $value : Value(value == "yes")
354         Input(name == "Air filter o.k.?" && values contains $value)
355         $solution : SolutionInput(name == "Clogged air filter")
356     then
357         $solution.setValue(N7);
358 end
359
360 rule "R26"
361     when
362         $value : Value(value == "no")
363         Input(name == "Air filter o.k.?" && values contains $value)
364         $solution : SolutionInput(name == "Clogged air filter")
365     then
366         $solution.setValue(P7);
367 end
368
369 rule "R27"
370     when
371         $value1 : Value(value == "weak acceleration")
372         Input(name == "Driving" && values.size > 0 && values not contains
            $value1)

```

```

373     $value2 : Value(value == "unsteady idle speed")
374     Input(name == "Driving" && values.size > 0 && values not contains
        $value2)
375     $solution : SolutionInput(name == "Clogged air filter")
376     then
377         $solution.setValue(N5);
378 end
379
380 rule "R28"
381     when
382         $value : Value(value == "unsteady idle speed")
383         Input(name == "Driving" && values contains $value)
384         $solution : SolutionInput(name == "Clogged air filter")
385     then
386         $solution.setValue(P4);
387 end
388
389 rule "R29"
390     when
391         $value : Value(value == "weak acceleration")
392         Input(name == "Driving" && values contains $value)
393         $solution : SolutionInput(name == "Clogged air filter")
394     then
395         $solution.setValue(P5);
396 end
397
398 rule "R30"
399     when
400         $value : Value(value == "turns over")
401         Input(name == "Starter" && values.size > 0 && values not contains
            $value)
402         $solution : SolutionInput(name == "Clogged air filter")
403     then
404         $solution.setValue(N6);
405 end
406
407 rule "R34"
408     when
409         $value : Value(value == "black")
410         Input(name == "Exhaust fumes" && values.size > 0 && values not
            contains $value)
411         $solution : SolutionInput(name == "Clogged air filter")
412     then
413         $solution.setValue(N5);
414 end
415
416 rule "R35"
417     when
418         $value : Value(value == "slightly increased")
419         Input(name == "Mileage evaluation" && values contains $value)
420         $solution : SolutionInput(name == "Clogged air filter")

```



```

421     then
422         $solution.setValue(P3);
423     end
424
425     rule "R36"
426     when
427         $value : Value(value == "normal")
428         Input(name == "Mileage evaluation" && values contains $value)
429         $solution : SolutionInput(name == "Clogged air filter")
430     then
431         $solution.setValue(N4);
432     end
433
434     rule "R37"
435     when
436         $value : Value(value == "increased")
437         Input(name == "Mileage evaluation" && values contains $value)
438         $solution : SolutionInput(name == "Clogged air filter")
439     then
440         $solution.setValue(P4);
441     end
442
443     rule "R47"
444     when
445         $value : Value(value == "yes")
446         Input(name == "Ignition timing o.k.?" && values contains $value)
447         $solution : SolutionInput(name == "Bad ignition timing")
448     then
449         $solution.setValue(N7);
450     end
451
452     rule "R48"
453     when
454         $value : Value(value == "no")
455         Input(name == "Ignition timing o.k.?" && values contains $value)
456         $solution : SolutionInput(name == "Bad ignition timing")
457     then
458         $solution.setValue(P7);
459     end
460
461     rule "R46"
462     when
463         $value : Value(value == "turns over")
464         Input(name == "Starter" && values.size > 0 && values not contains
465             $value)
466         $solution : SolutionInput(name == "Bad ignition timing")
467     then
468         $solution.setValue(N6);
469     end
470
471     rule "R49_1_2"

```

```

471     when
472         $value1 : Value(value == "knocking")
473         Input(name == "Engine noises" && values contains $value1)
474         $solution : SolutionInput(name == "Bad ignition timing")
475     then
476         $solution.setValue(P5);
477 end
478
479 rule "R49_2_2"
480     when
481         $value2 : Value(value == "ringing")
482         Input(name == "Engine noises" && values contains $value2)
483         $solution : SolutionInput(name == "Bad ignition timing")
484     then
485         $solution.setValue(P5);
486 end
487
488 rule "R50"
489     when
490         $value1 : Value(value == "knocking")
491         Input(name == "Engine noises" && values.size > 0 && values not
            contains $value1)
492         $value2 : Value(value == "ringing")
493         Input(name == "Engine noises" && values.size > 0 && values not
            contains $value2)
494         $solution : SolutionInput(name == "Bad ignition timing")
495     then
496         $solution.setValue(N3);
497 end
498
499 rule "R42"
500     when
501         $value1 : Value(value == "weak acceleration")
502         Input(name == "Driving" && values.size > 0 && values not contains
            $value1)
503         $value3 : Value(value == "unsteady idle speed")
504         Input(name == "Driving" && values.size > 0 && values not contains
            $value3)
505         $value4 : Value(value == "delayed take-off")
506         Input(name == "Driving" && values.size > 0 && values not contains
            $value4)
507         $solution : SolutionInput(name == "Bad ignition timing")
508     then
509         $solution.setValue(N4);
510 end
511
512 rule "R43"
513     when
514         $value : Value(value == "unsteady idle speed")
515         Input(name == "Driving" && values contains $value)
516         $solution : SolutionInput(name == "Bad ignition timing")

```

```

517     then
518         $solution.setValue(P1);
519     end
520
521     rule "R44"
522     when
523         $value : Value(value == "delayed take-off")
524         Input(name == "Driving" && values contains $value)
525         $solution : SolutionInput(name == "Bad ignition timing")
526     then
527         $solution.setValue(P3);
528     end
529
530     rule "R45"
531     when
532         $value : Value(value == "weak acceleration")
533         Input(name == "Driving" && values contains $value)
534         $solution : SolutionInput(name == "Bad ignition timing")
535     then
536         $solution.setValue(P4);
537     end
538
539     rule "R1"
540     when
541         $value1 : Value(value == "sooty black")
542         Input(name == "Exhaust pipe color" && values contains $value1)
543         $value2 : Value(value == "unleaded gasoline")
544         Input(name == "Fuel" && values contains $value2)
545         $input : ChoiceInput(name == "Exhaust pipe color evaluation")
546     then
547         $input.setValue("abnormal");
548     end
549
550     rule "R2"
551     when
552         $value1 : Value(value == "sooty black")
553         Input(name == "Exhaust pipe color" && values contains $value1)
554         $value2 : Value(value == "diesel")
555         Input(name == "Fuel" && values contains $value2)
556         $input : ChoiceInput(name == "Exhaust pipe color evaluation")
557     then
558         $input.setValue("normal");
559     end
560
561     rule "R4"
562     when
563         Input(name == "Num. Mileage evaluation" && numValue <= 130.0)
564         Input(name == "Num. Mileage evaluation" && numValue >= 110.0)
565         $input : ChoiceInput(name == "Mileage evaluation")
566     then
567         $input.setValue("slightly increased");

```

```

568 end
569
570 rule "R5"
571     when
572         Input(name == "Num. Mileage evaluation" && numValue > 130.0)
573         $input : ChoiceInput(name == "Mileage evaluation")
574     then
575         $input.setValue("increased");
576     end
577
578 rule "R6"
579     when
580         Input(name == "Num. Mileage evaluation" && numValue > 0 &&
581             numValue < 110.0)
582         $input : ChoiceInput(name == "Mileage evaluation")
583     then
584         $input.setValue("normal");
585     end
586
587 rule "R3_1_2"
588     when
589         $value1 : Value(value == "light grey")
590         Input(name == "Exhaust pipe color" && values contains $value1)
591         $input : ChoiceInput(name == "Exhaust pipe color evaluation")
592     then
593         $input.setValue("normal");
594     end
595
596 rule "R3_2_2_3_2"
597     when
598         $value3 : Value(value == "grey")
599         Input(name == "Exhaust pipe color" && values contains $value3)
600         $input : ChoiceInput(name == "Exhaust pipe color evaluation")
601     then
602         $input.setValue("normal");
603     end
604
605 rule "R3_2_2_4_2"
606     when
607         $value4 : Value(value == "brown")
608         Input(name == "Exhaust pipe color" && values contains $value4)
609         $input : ChoiceInput(name == "Exhaust pipe color evaluation")
610     then
611         $input.setValue("normal");
612     end
613
614 %
615 %%DroolsSession
616 get Clogged air filter
617 set Exhaust fumes = black
618 set Fuel = unleaded gasoline

```

```
618 set Engine start = engine barely starts
619 fire
620 get Clogged air filter
621
622 @Name: CloggedAirFilter_Suggested
623 %
```